

## ARGUMENTS / REMARKS

### Status of Claims

Claims 1-25 were previously pending in the present application. With the present amendment Claims 19-25 are cancelled and new Claims 26-31 are added.

### Rejection under 35 U.S.C. § 102(e)

Claims 1-25 were rejected pursuant to 35 U.S.C. § 102(e), as being anticipated by U.S. Patent No. 6,804,818 issued to Codella et al. ("Codella"). Applicant respectfully traverses the Examiner's rejection, and submits that none of the pending claims are anticipated by Codella. Codella fails to disclose, either explicitly or inherently, each and every limitation of the pending claims, as required for a rejection under § 102(e).

For example, it is respectfully submitted that Codella fails to teach the limitation in independent Claim 1: "*wherein a second distributed application component is identified in said request as a recipient of said request*".

Similarly, Codella fails to teach the limitation in independent Claims 27 and 29: "*wherein the message identifies a second distributed application component*".

Likewise, Codella fails to teach the limitation in Claim 3: *wherein said recipient is identified by a second property of said second distributed application component included within said request*"; and the limitation in Claim 4: "*wherein said second property is a unique identifier of said second distributed application component*".

Codella teaches mechanisms for sending and receiving anonymous invocations between message beans. Thus, in Codella, the messages are anonymous and do not identify either the sender or the intended recipient of the invocation (*see, e.g.*, Codella, abstract; col. 3 line 61 to col. 4 line 15).

In clear contrast, each of the pending claims requires that the message sent between components, identify the recipient of the message.

In the Office Action, with respect to Claim 3, the Examiner contends that Codella teaches “an ‘Operation’ field, [...as] a property included within the request that identifies the message bean that implements the requested interface” (*see*, Office Action p. 4, *referring to*, Codella, col. 10 line 51 to col. 11 line 15).

Applicant respectfully disagrees. The “Operation” field in a message does not identify an intended recipient of the message. Rather, the “Operation” field is used to convey “a specific request of a receiver, namely to perform the specified operation” (col. 10, line 65-67). For example, the “Operation” field may be formatted as “[Operation = {“debit”},AccountId=long,Amount=double]” (col. 10 line 56). This is obviously not an identifier of a recipient, but merely an identifier of the desired task to be performed.

In fact, the very concept of a message that specifically identifies a recipient is contrary to the very essence of Codella’s invention. For example, Codella touts the advantage of the invention as follows:

“The integration mechanism permits an object oriented component, heretofore referred as a message bean, to perform anonymous invocations that are serviced by other message beans or by message-oriented servers in such a way the requesting message bean is unaware of whether the server of the anonymous invocation is either a message bean or a message oriented server. (emphasis added)

Codella, col. 3, lines 61-67.

Thus clearly, Codella teaches away from a requesting message bean that identifies within an invocation (i.e. message) the actual message bean that is to service the invocation.

With respect to Claim 4, the Examiner contends that Codella discloses that a container “will extract the corresponding field from an incoming message and use it to identify the instance of the message bean to invoke”. (*see*, Office Action p. 4, *citing*, Codella, col. 13, lines 15-17).

But the fact that a container of a message bean can extract a field from a received message and use it to identify an instance of the message bean to invoke, does not mean that the message identifies the actual message bean.\*

Thus, as noted, Codella fails to teach – and in fact teach away from – a message that identifies the component that is the intended recipient of the message.

In the same vein, Codella fails to teach the step in Claim 1 of: *"identifying by the middleware program a publish/subscribe topic by identifying a first property of said second distributed application component"*.

In the Office Action the Examiner contends that the Codella teaches that the output port of a message proxy of a first message bean specifies a destination; that the input port of second message bean corresponds to the same destination; and that Codella further teaches that the destination is a publish/subscribe topic (*see*, Office Action, pp. 2-3).

The Examiner apparently broadly defines the term "property" in Claim 1 to include the input port of the second message bean.

But even if we were to accept the Examiner's broad definition of "property" to include the input port of the second message bean, Codella however does not teach that the message proxy of the first message bean identifies the destination by identifying the input port of the second message bean. Clearly, the message proxy of the first message bean cannot possibly identify the destination by looking to the input port of a second message bean. In fact, as noted, according to Codella the second message bean is unknown to the first message bean, thus it would be impossible for the first message to identify the input port of the second message bean.

In clear contrast, Claim 1 requires that the publish/subscribe topic is identified by the middleware program in order to publish the request of the first component, by first identifying a property of the second component. For example, the middleware program

---

\* Much in the same way as a bottle of medicine that says "keep refrigerated" does not mean that the bottle identifies the address or name of a house in which a refrigerator is located.

may determine that the second property is a database and hence identify the publish/subscribe topic referred to as “database.reply” (*see e.g.*, specification at ¶ 0033).

Similarly, Claims 27 and 29 require that the message is published to a publish/subscribe topic identified based on a property of the second component. This requirement would of course be impossible in Codella where the second message bean (component) is unknown to the first message bean (component). Thus, Codella cannot anticipate Claims 27 and 29.

For the same reasons as noted above, the other dependent claims are not anticipated. As such, it is respectfully submitted that Codella does not anticipate any of the pending claims.

**CONCLUSION**

In view of the present amendment to the claims and the foregoing remarks, it is respectfully requested that the outstanding rejection be withdrawn and a notice of allowance be issued for the present application.

In the event the Examiner believes that an interview would be helpful in advancing the present application, the Examiner is respectfully requested to contact the undersigned at the number indicated below.

Dated: April 5, 2009

Respectfully submitted,

**INCEPTIA LLC (Assignee) \***

Tel.: (718) 928-2213

By: *Benzion Wachsman*

Name: *Benzion Wachsman*

Title: *General Manager*

\* A statement under 37 C.F.R. § 3.73(b) is attached